Comparing ML classifiers with average accuracy and execution time

2021025205 Tae Wook Kang

1. Abstraction

This report is about an experiment that compares several ML models' accuracy and performance by using the MNIST dataset. Through a scikit-learn library, it was able to repeat the process of training each model and taking a result. The results of this experiment show that non-linear SVM and k-NN are the most effective and accurate algorithms.

2. Problem setting

a. *MNIST datasets*

The MNIST database is a large database of hand-written digits. The digits dataset consists of 8x8 pixel images of digits. The images attribute of the dataset stores 8x8 arrays of grayscale values for each image.

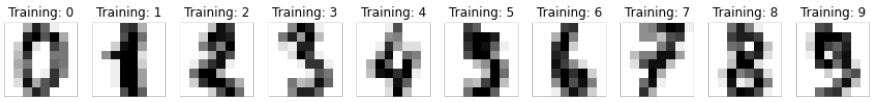


Figure 1 MNIST datasets

There are 1797 images in this database, and each image has 64 attributes as it is an 8x8 pixel scale image. Each pixel has an integer between 0 to 16, which represents the contrast of the pixel. Thus, each number can be expressed as a matrix.

```
[[ 0.  0.  0. 12. 13.  5.  0.  0.]
 [ 0.  0.  0. 11. 16.  9.  0.  0.]
 [ 0.  0.  3. 15. 16.  6.  0.  0.]
 [ 0.  7. 15. 16. 16.  2.  0.  0.]
 [ 0.  0.  1. 16. 16.  3.  0.  0.]
 [ 0.  0.  1. 16. 16.  6.  0.  0.]
 [ 0.  0.  1. 16. 16.  6.  0.  0.]
 [ 0.  0.  0. 11. 16. 10.  0.  0.]]
```

Figure 2 '0' represented as a matrix of numbers

Since there are 1797 8 x 8 matrices of digit images, these data are represented in 3 dimensions (1797 x 8 x 8). In this experiment, I transformed the data into 2 dimensions because models that I wanted to train through scikit-learn accept two-dimensional data (1797 x 64). The target data, which are the answer, are one-dimensional (1797 x 1). Because of the small number of data, this dataset allows us to find algorithms that perform well even with small amounts of data.

b. *The purpose of experiments*

The purpose of this experiment is to identify the best algorithm for classifying values of hand-written digits correctly by comparing the accuracy and performance of each model. To measure the accuracy of the model, I compared data predicted by a model and target data. I got data on performance by measuring the time spent on learning and predicting.

3. Experiments

a. *Algorithms used in this experiment*

In this experiment, I have selected nine classification models.

i. Logistic Regression

Logistic regression is a supervised learning algorithm that uses regression to predict the probability that data belongs to the category from 0 to 1 and categorizes it into a more likely category based on that probability.

ii. Stochastic Gradient Descent (SGD)

SGD is an iterative method for optimizing an objective function with small mini-batches. Strictly speaking, SGD is merely an optimization technique. SGD's model is equal to the logistic regression model.

iii. Linear Discriminant Analysis (LDA or FDA)

LDA is a method to find a linear combination of features that characterizes or separates two or more classes of objects or events. LDA model fits a Gaussian density to each class, assuming that all classes share the same covariance matrix. It makes a model by maximizing the difference in mean versus variance.

iv. Multi-Layer Perceptron (MLP)

MLP is a neural network in which one or more intermediate layers exist between the input layer and the output layer. It adjusts its parameters effectively through backpropagation.

v. Support Vector Machine (SVM) – Linear SVM and Non-linear SVM

SVM is a classifier that finds the decision boundary as far away as possible from two classes by maximizing the margin. It aims to classify classes while satisfying certain conditions. SVM can be divided into linear SVM and non-linear SVM by the type of kernel. In this experiment, I used linear kernel to linear SVM model and gaussian kernel to non-linear SVM model.

vi. Naive Bayes – Gaussian Naive Bayes and Multinomial Naive Bayes

Naive Bayes is a set of supervised learning algorithms based on applying Bayes' theorem with the naive assumption of conditional independence between every pair of features given the value of the class variable. In this experiment, I used Gaussian and Multinomial Naive Bayes models.

vii. k-Nearest Neighbors (k-NN)

k-NN algorithm predicts the class of a query point based on the information of the k number of nearest neighbors. It does not require an explicit training process as it makes predictions by finding the distance between the query point and training data.

b. Data processing - Splitting dataset

To train the models and test them, I divided the whole data into two parts, training data and test data. In these experiments, I shuffled whole data of images before splitting them and split training data and test data at the ratio of 8:2.

c. Learn each model

This experiment was conducted using scikit-learn library. The parameters of the model, except logistic regression, were set as the default parameter values of the model. To make the model converge, I modified a *max_iter* parameter of the logistic regression model to 5000.

All models have been fitted 1000 times in total. Before each iteration, the data processing process is conducted by shuffling data and splitting them into training data and test data. Each iteration is accomplished by all algorithms learning the same training data set, measuring the time spent in fitting and predicting process, and measuring accuracy through the same test data set.

d. Get a result

Accuracy, time spent to learn, and time spent to predict are stored in each matrix. Those are (the number of iterations) x (the number of algorithms) matrix, so the average estimated value of each model can be obtained by taking the values of each column.

```
for j in range(len(names)):
  accuracy = 0

  for i in range(iteration):
    accuracy += eachAccuracy[i][j]

  print(str(accuracy / iteration) + " - " + names[j])
```

Figure 3 Code used to obtain average accuracy

By comparing each model's average accuracy, average time spent in learning, and average time spent in predicting, the best algorithm is chosen. For the first iteration, the program conducting this experiment displays a classification report and a confusion matrix. From these data, more detailed prediction results of each model can be found.

4. Experimental results

a. Experimental results table

Table 1 Average accuracy (round to 6 digits)

Logistic Regression	SGD	LDA	MLP	Linear SVM	Non-linear SVM	Gaussian Naive Bayes	Multinomial Naive Bayes	k-NN
0.96400	0.94848	0.95275	0.97483	0.97899	0.98746	0.83974	0.89914	0.98589

Table 2 Average time spent to learn (seconds, round to 6 digits)

Logistic Regression	SGD	LDA	MLP	Linear SVM	Non-linear SVM	Gaussian Naive Bayes	Multinomial Naive Bayes	k-NN
6.41514	0.08395	0.02945	1.88705	0.04657	0.74129	0.00594	0.00780	0.00093

Table 3 Average time spent to predict (seconds, round to 6 digits)

Logistic Regression	SGD	LDA	MLP	Linear SVM	Non-linear SVM	Gaussian Naive Bayes	Multinomial Naive Bayes	k-NN
0.00036	0.00044	0.00038	0.00157	0.01048	0.02435	0.00129	0.00029	0.04485

b. Analysis on the accuracies and performances, Pros and Cons of each algorithm.

i. Logistic Regression

The learning time of Logistic Regression, which was 6.41514 seconds, was the longest among the algorithms. This result appears because this model repeats calculations until its objective function converges. The average accuracy of this model was about 96.4%, which was the middle figure among the models. The prediction time of the Logistic Regression model had the second-best value among the algorithms because its prediction ended up with a simple calculation of values.

Pros: It is the simplest machine learning algorithm, so it is easy to implement.

Cons: Learning takes a long time if the model does not converge properly.

ii. Stochastic Gradient Descent

Since the model is iteratively learned with small batches of data, the time spent learning in SGD is more shorter than that of Logistic Regression. However, the average accuracy of the model is about 95%, which is slightly low because it is not learned by the

entire training dataset. SGD model had a good value on time spent predicting since it is basically a Logistic Regression model.

Pros: It is more effective than Logistic Regression when a dataset is large.

Cons: It can converge at local minimum and saddle point.

iii. Linear Discriminant Analysis

LDA models reflect the covariance structure between data, while Naive Bayes models do not. For this reason, these models showed about 5 to 12 percent higher average accuracy than Naive Bayes models. This algorithm model took longer time to learn than Naive Bayes models, but it is faster than other linear models.

Pros: It is more common than Naive Bayes as it considers a covariance of data.

Cons: It can be inaccurate since it assumes the normal distribution of the data.

iv. Multi-Layer Perceptron

The MLP model recorded a good fourth-place accuracy, which was 97.5% since it adjusts its parameters through multiple layers in the learning process. Time spent learning was a bad result of being second. The reason the model made a bad learning time is that this model adjusts its parameters through backpropagation, in which multiple layers of learning occur. Because of the multilayer, this model showed a prediction time of about four times longer than the single-layer models such as Logistic Regression, SGD, and LDA models.

Pros: Since it has multiple layers, it can create complex nonlinear models.

Cons: Learning takes a long time because of the large number of parameters.

v. Support Vector Machine

Linear SVM and Non-linear (RBF) SVM models respectively showed an accuracy of 97.9% and 98.7%, which is third and first place. It means that SVM can classify the data well even with small training data. Non-linear SVM, which uses a gaussian kernel, recorded better accuracy than Linear SVM since the gaussian kernel can divide complex data by increasing a dimension. However, since the dataset has lots of classes to classify, SVM models need more time to build a model. Especially, Non-linear SVM needs more calculation than linear SVM and other algorithms such as Logistic Regression and LDA, it spent about 0.74 seconds in learning and 0.024 seconds in predicting process.

Pros: It shows good predictions even with a small amount of data.

Cons: As the dimension of the data grows, learning takes a long time.

vi. Naive Bayes

Gaussian Naive Bayes and Multinomial Naive Bayes models recorded low average accuracy, which was 84% and 90% because they learn data in anticipation of probability independence. The Multinomial Naive Bayes model showed better results than the Gaussian Naive Bayes model since the experiment's problem is to infer the value of discrete hand-written numbers.

Pros: It can classify the class of input data fast.

Cons: Algorithms' accuracy is low if the data are not mutually independent.

vii. k-Nearest Neighbors

As the k-NN model uses all the data for training while classifying a new data point, it showed a precise average accuracy, about 98.6%. The learning took only 0.00093 seconds because the k-NN model does not need a learning process. However, it spent 0.04485 seconds in predicting, which is the largest value among the result. It is

because the model calculates the distance between training data and a query point in predicting phase.

Pros: It is simple and efficient since there does not involve a learning process.

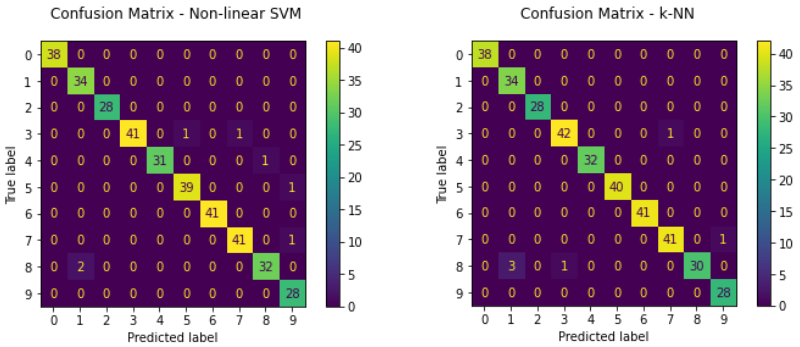
Cons: Prediction slows down if there are a lot of data to compare with a query.

c. Comparison between the accuracies of the algorithms.

In terms of average accuracy, the Non-linear SVM algorithm and k-NN algorithm are the best algorithms. Unlike linear models such as Logistic Regression, LDA, and linear SVM, Non-linear SVM performs better because it can create a non-linear decision boundary by expanding the data's dimension. The k-NN algorithm makes better results in distinguishing query points' value than other models because k-NN uses actual data to classify the query point, while other algorithms use the calculated model which is trained to discriminate input data. Naive Bayes models recorded lower accuracy than other models like LDA. As these models simply assume that the data are mutually independent, they made a bad result. On the other hand, LDA showed better accuracy than Naive Bayes models because it concerned covariance between data.

5. Discussion

When comparing the accuracy and performance of each model, Non-linear SVM and k-NN are the most appropriate algorithm that can effectively distinguish numbers even with small numbers of data. However, in detail, both algorithms respectively showed 94% and 92% accuracy to distinguish '1'.



As can be seen from the confusion matrix above, the models confused '1' and '8'. This seems to be because both numbers are vertically long and in a symmetrical form, so it is not easy to distinguish features between numbers. The Linear Regression model, first covered in this class, was not used in the experiment because it is not a classification algorithm. This experiment was conducted simply with few data and basic parameters of the model. In the next experiment, I want to compare algorithms with large-scale datasets and the detailed adjustment of parameters.

6. Conclusion

Non-linear SVM and k-NN were the best classifiers for hand-written digits, with an accuracy of about 98%. Since the k-NN model showed a shorter time in the learning and predicting phase, it is the most accurate and effective model on a small dataset among the models used in the experiment. The k-NN model showed good accuracy because it used real data to guess the value of the query point. Among algorithms that get a discrimination model through learning, Non-linear SVM showed the best results because its decision boundaries are non-linear. Naive Bayes models were fast at learning, but their accuracy was low because they assumed mutual independence of data. On the other hand, the LDA model had higher accuracy than them as it concerns dependent states of data. MLP was slow at learning due to the large number of parameters that should be calculated. Logistic Regression took a lot of time in the convergence process. SGD, designed to solve that problem, was fast but relatively less accurate than the Logistic Regression model.